■ ปราโมทย์ พลังสันติกุล

SOFT MAIL

เรียงลำดับข้อมูลธรรมดาที่ไม่ธรรมดาด้วยกราฟบาร์

เรียงลำดับข้อมูลด้วยเทคนิค Bubble Sort, Selection Sort และ Quick Sort เพื่อการเรียงลำดับข้อมูลที่รวดเร็วในแบบธรรมดาที่ไม่ธรรมดา

การเรียงลำดับข้อมูล โดยมากแล้วนิยมที่จะนำตัวเลขมาหนึ่งชุด จากนั้นจะนำมาจัดเรียง ซึ่งจะเห็นได้ทั่วไป แต่ถ้าหากเราต้องการที่จะดูขั้นตอนการจัดเรียงลำดับข้อมูล การสลับตำแหน่งของตัวเลขเพื่อการเรียงลำดับข้อมูลอาจจะทำให้เราเกิดอาการมึนไปกับตัวเลขได้ และหากมีจำนวนตัวเลขมาก ๆ ด้วยแล้ว เรียกว่าไม่ต้องกล่าวถึงเลยครับ เพราะฉะนั้นเรามาดูขั้นตอนการจัดเรียงข้อมูลแบบใหม่ด้วยวิธีการธรรมดาที่ไม่ธรรมดาแล้วไม่ธรรมดาอย่างไร ที่ไม่ธรรมดาคือ เราจะใช้สัญลักษณ์กราฟบาร์ ในการแสดงการจัดเรียงลำดับข้อมูลกันครับ (อาจจะเป็นเรื่องธรรมดาสำหรับท่านผู้อ่าน แต่สำหรับผู้เขียนเป็นการจัดเรียงข้อมูลธรรมดาที่ไม่ธรรมดาครับ)

การเรียงลำดับข้อมูล (DATA SORT) ถือเป็นงานพื้นฐานส่วนหนึ่งของการจัดการข้อมูล และในการศึกษาการเขียนโปรแกรมในหลักสูตรจะมีเนื้อหาเกี่ยวกับการเรียงลำดับข้อมูลอยู่ด้วย ซึ่งเป็นโปรแกรมพื้นฐานที่นักพัฒนางานทางคอมพิวเตอร์จะต้องสามารถเขียนได้ (การเรียงลำดับข้อมูลเป็นส่วนหนึ่งของการทดสอบความรู้ทางด้านการเขียนโปรแกรมที่ยอดฮิตเลยทีเดียว) โดยปกติภาษาโปรแกรมต่าง ๆ โดยมากแล้วจะมีฟังก์ชันการเรียงลำดับข้อมูลให้มา เช่น ในภาษาซี จะมี Quick Sort มาให้ เป็นต้น หรือหากเป็นภาษาการเขียนโปรแกรมประเภทอื่นมักจะแฝงมาในรูปแบบอื่นที่ไม่ใช่เป็นฟังก์ชันที่ให้เรียกใช้ได้เลย แล้วใน Visual BASIC จะมีหรือไม่ มีครับ แต่แฝงมากับ Control ActiveX (OCX) และการเรียกใช้งานทางฐานข้อมูลนั้นแสดงได้ว่าหากคุณจะใช้คุณสมบัติของการเรียงลำดับข้อมูล คุณก็จะต้องใช้ Control ActiveX หรือใช้งานฐานข้อมูลด้วย โดยที่เราไม่ได้มีความจำเป็นต้องใช้งาน Control นั้น ๆ เลย (ผู้เขียนหาไม่พบว่าใน Visual BASIC มีฟังก์ชันการเรียงลำดับข้อมูล)

จึงเป็นที่มาของการสร้างฟังก์ชันการเรียงลำดับข้อมูลเพื่อนำมาใช้กับงานที่ได้พัฒนาขึ้นมาโดยไม่ต้องอาศัย Control ActiveX สามารถที่จะเรียงลำดับข้อมูลได้ และเป็นฟังก์ชันการเรียงลำดับข้อมูลทำให้สามารถนำไปเพิ่มในซอร์สโค้ดของเราและเรียกใช้ได้เลยในทันที (ง่ายไหมครับ)

ในการสร้างโค้ดการเรียงลำดับข้อมูลนั้นผู้เขียนจะนำเสนอการเรียงลำดับข้อมูล 3 รูปแบบ คือ

1. BUBBLE SORT
2. SELECTION SORT
3. QUICK SORT

เราจะมาดูกันครับว่าในการจัดเรียงข้อมูลด้วยเทคนิคทั้งสามว่าวิธีการใดเร็วที่สุดในการจัดเรียงข้อมูล

BUBBLE SORT

เป็นหลักการเรียงข้อมูล โดยการนำข้อมูลที่อยู่ติดกันมาเปรียบเทียบกันทีละคู่ และหากข้อมูลที่เปรียบเทียบข้อมูลใดน้อยกว่าจะอยู่ทางซ้ายและจะเปรียบเทียบเป็นคู่ ๆ ไปจนหมดข้อมูล เมื่อทำจนหมดข้อมูลจะทำให้ข้อมูลที่น้อยสุดอยู่ทางซ้ายของข้อมูล ลักษณะอัลกอริธึมของ Bubble Sort ที่เขียนเป็นโค้ดภาษาเบสิกได้ดังนี้

```
For I = Ahigh To 0 Step -1
    For J = 0 To Ahigh - 1
If A(J) > A(J + 1) Then
T = A(J)
A(J) = A(J + 1)
A(J + 1) = T
End If
    Next J
Next I
```

SELECTION SORT

เป็นหลักการเรียงข้อมูล โดยการนำข้อมูลตัวที่น้อยที่สุดมาเก็บไว้ก่อน โดยการหาทั้งหมดของข้อมูลหากพบตัวไหนมีค่าน้อยกว่าก็นำมาเก็บไว้ภายนอกและกลับไปค้นหาข้อมูลต่อที่เหลือทั้งหมด และนำตัวที่น้อยที่สุดมาเก็บไว้เช่นเดียวกันจนหมดข้อมูล เราก็จะได้ข้อมูลจากน้อยไปหามาก ลักษณะอัลกอริธึมของ Selection Sort ที่เขียนเป็นโค้ดภาษาเบสิกได้ดังนี้

```
For I = 0 To Ahigh - 1
 For J = Ahigh To I + 1 Step -1
    If A(I) > A(J) Then
T = A(I)
A(I) = A(J)
A(J) = T
 End If
    Next J
    Next I
```

QUICK SORT

เป็นหลักการเรียงข้อมูล โดยการใช้วิธีการแบ่งข้อมูลออกเป็น 2 ส่วน และจะทำการเรียกตัวเองเพื่อเรียงลำดับข้อมูลทั้งสองส่วน ในหลักการของวิธีนี้จะใช้หลักการของการเรียกใช้ตนเองที่เรียกว่า Recursive (Recursive : เป็นหลักการทำงานที่ให้ฟังก์ชันหรือซับฟังก์ชันเรียกตนเองมาทำงานต่อ) ลักษณะอัลกอริธึมของ Quick Sort ที่เขียนเป็นโค้ดภาษาเบสิกได้ดังนี้

```
Lo = iLo
 Hi = iHi
 intMid = A((Lo + Hi) / 2)
 Do
 While (A(Lo) < intMid)
  Lo = Lo + 1
 Wend
 While (A(Hi) > intMid)
  Hi = Hi - 1
 Wend
 If Lo <= Hi Then
  T = A(Lo)
  A(Lo) = A(Hi)
  A(Hi) = T
  Lo = Lo + 1
  Hi = Hi - 1
 End If
 Loop While (Lo <= Hi)
 If Hi > iLo Then Call VB_QuickSort(A, iLo, Hi)
 If Lo < iHi Then Call VB_QuickSort(A, Lo, iHi)
```

อธิบายโปรแกรมก่อนทดสอบการใช้งาน

กำหนดตัวแปรเพื่อใช้ในโปรแกรม

Dim LineX(70) As Double, LineY(70) As Double ใช้ในการเก็บค่าที่ได้จากการสุ่มตัวเลข

Dim tmpLineX(70) As Double ใช้ในการเก็บค่าที่ได้จากการสุ่มตัวเลขเพื่อนำกับมาใหม่โดยต้องการใช้ตัวเลขสุ่มค่าเดิม

Private Sub RandLineBar()

โพรซีเดอร์ RandLineBar ใช้ในการสุ่มค่าตัวเลขเพื่อใช้ในการสร้างกราฟบาร์ โดยตัวแปรอะเรย์ LineY(I) ใช้เก็บช่วงห่างของกราฟบาร์แต่ละแท่งในแนวนอน และตัวแปรอะเรย์ LineX(I) ใช้เก็บขนาดของกราฟบาร์ที่จะแสดงว่าแต่ละแท่งมีขนาดเท่าไร

```
Randomize   'ฟังก์ชันที่ใช้ในการสุ่มเลขค่า
LY = 680    'กำหนดตำแหน่งเริ่มต้นของแกน Y
LX = 1000   'กำหนดตำแหน่งเริ่มต้นของแกน X
For i = 1 To 70
    LY = LY + 50  'เพิ่มค่าแนวแกน Y ทุก ๆ ห้าสิบ เพื่อให้กราฟบาร์
                  แต่ละแท่งห่างกัน 'ตำแหน่งแท่งละห้าสิบ
    LineY(i) = LY  'นำค่าในแนวแกน Y ไปเก็บไว้ในตัวแปรอะเรย์
                   เพื่อใช้ในแนวแกน Y
NumRnd = Int((1000 * Rnd) + 10 * i)  'กำหนดเลขสุ่มให้กับแกน X
LineX(i) = LX + NumRnd 'ขนาดเริ่มต้นของแกน X บวกกับค่าสุ่มที่ได้
tmpLineX(i) = LineX(i)  'เก็บค่าเดิมไว้หากต้องการใช้ค่าเดิมล่าสุด
                        ในการเรียงลำดับ
Next
```

ในโพรซีเดอร์นี้จะมีการใช้ฟังก์ชันในการสุ่มเลข โดยต้องกำหนดเพื่อให้มีการใช้งานฟังก์ชันสุ่มตัวเลขด้วยฟังก์ชัน Randomize และใช้ฟังก์ชัน Int((1000 * Rnd) + 10 * i)

ในการสุ่มเลขโดยค่าที่สุ่มจะอยู่ในช่วง ตั้งแต่ 10 * I จนถึง 1,000 เช่น หากต้องการให้มีเลขสุ่มอยู่ในช่วง 1 – 20 จะกำหนดดังได้นี้

```
Int((20 * Rnd) + 1)
```

Private Sub cmdOldData_Click()

โพรซีเดอร์ที่ใช้ในการนำค่าของเลขสุ่มที่เก็บไว้นำกลับมาใช้ใหม่โดยไม่ต้องการสุ่มค่าใหม่ ให้กับตัวแปรที่ใช้ในการทำงานในการเรียงลำดับข้อมูล

```
For i = 1 To 70
  LineX(i) = tmpLineX(i) 'นำค่าเลขสุ่มเก่ากลับมาใช้
Next
```

Private Sub PrintLineSwap (ByRef LBar() As Double, ByVal PXO As Integer)

โพรซีเดอร์ที่ใช้ในการลบกราฟบาร์ในตำแหน่งที่จะมีการสลับค่ากับค่าใหม่ โดยการลบค่ากราฟบาร์นั้นจะอาศัยหลักการเขียนค่าสีทับค่าสีเดิมให้กับกราฟบาร์โดยให้มีสีเดียวกับสีพื้น

```
If Not IsEmpty(PXO) Then
Line (200, LineY(PXO))-(LBar(PXO), LineY(PXO)),
QBColor(7)
End If
```

โดยการใช้ฟังก์ชัน QBColor(7) โดยเลข 7 จะแสดงค่าของสีเทาซึ่งเป็นสีเดียวกับสีพื้น

Private Sub ShowLineBar()

โพรซีเดอร์ ที่ใช้ในการวาดรูปกราฟบาร์ ที่ได้จากการสุ่มค่าโดยอาศัยฟังก์ชัน Line ของ Visual BASIC

```
For i = 1 To 70
 DoEvents
 Line (200, LineY(i))-(LineX(i), LineY(i)), QBColor(12)
Next
```

การทดสอบและตารางเปรียบเทียบความเร็วของการเรียงลำดับข้อมูล

โปรแกรมการเรียงลำดับข้อมูลแสดงดังรูปที่ 1 เมื่อ RUN โปรแกรมจะแสดงรูปกราฟบาร์และพร้อมที่จะทำงานทันที โดยมีปุ่มใช้งานดังนี้

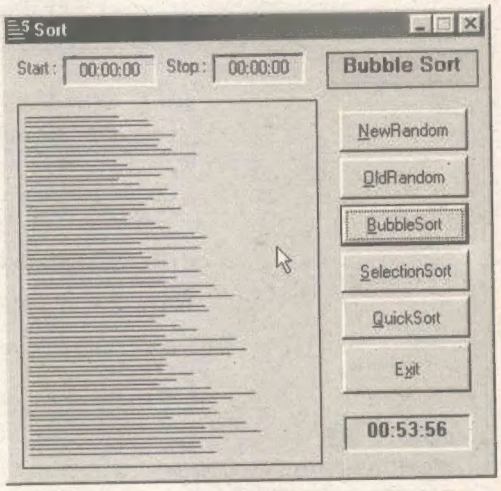
1. ปุ่มคำสั่ง NewRandDom ใช้ในการสุ่มกราฟบาร์ใหม่

2. ปุ่มคำสั่ง OldRandDom ใช้ในการนำค่าสุ่มล่าสุดกลับมาใหม่ (นำกราฟบาร์เก่ากลับมา)

3. ปุ่มคำสั่ง Bubble Sort ใช้ในการเรียงลำดับข้อมูลด้วยเทคนิค Bubble Sort

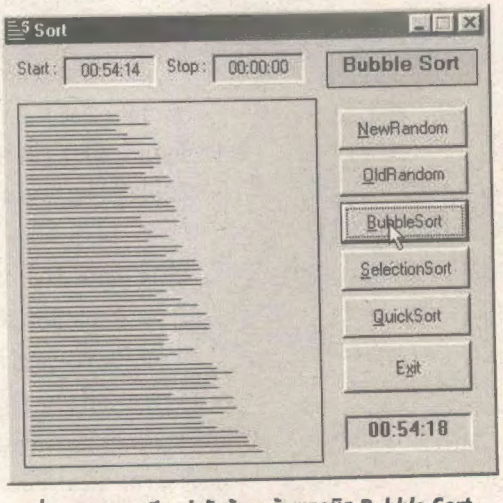
4. ปุ่มคำสั่ง Selection Sort ใช้ในการเรียงลำดับข้อมูลด้วยเทคนิค Selection Sort

5. ปุ่มคำสั่ง Quick Sort ใช้ในการเรียงลำดับข้อมูลด้วยเทคนิค Quick Sort

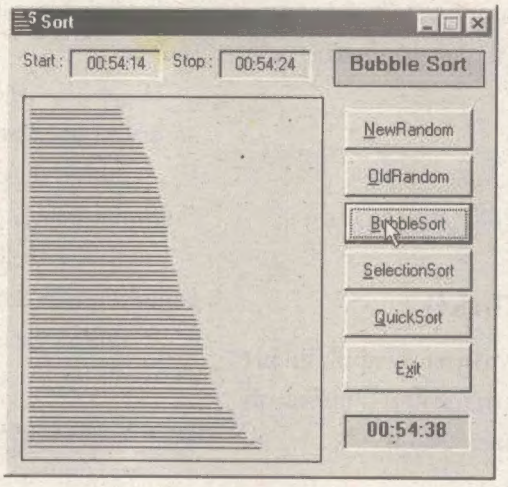


รูปที่ 1 โปรแกรม Sort โดยยังไม่ได้จัดเรียงข้อมูล

การทดสอบการทำงานโดยการเรียงลำดับข้อมูลโดยการใช้เทคนิค Bubble Sort โดยแสดงขั้นตอนการเรียงลำดับข้อมูลดังรูปที่ 2 และได้ผลลัพธ์แสดงดังรูปที่ 3 จะเห็นได้ว่าเริ่มต้นการเรียงลำดับข้อมูลที่เวลา 00:54:14 และสิ้นสุดการเรียงลำดับข้อมูลที่เวลา 00:54:24 โดยใช้เวลาทั้งสิ้นเท่ากับ 10 วินาที โดยประมาณ



รูปที่ 2 แสดงการเรียงลำดับข้อมูลด้วยเทคนิค Bubble Sort



รูปที่ 3 แสดงผลลัพธ์การเรียงลำดับข้อมูลด้วยเทคนิค Bubble Sort

หลังจากที่ได้ทำการทดสอบด้วยเทคนิคทั้ง 3 แล้ว ปรากฏว่าการเรียงลำดับข้อมูลด้วยเทคนิค Quick Sort จะสามารถทำการจัดเรียงข้อมูลได้เร็วที่สุด เรียงลำดับระยะเวลาที่ใช้และทดสอบแสดงได้ดังตารางที่ 1

ตารางที่ 1
แสดงการทดสอบการเรียงลำดับข้อมูล

การเรียงลำดับข้อมูลด้วยวิธี	เวลาเริ่ม	เวลาสิ้นสุด	ใช้เวลาทั้งหมด (วินาที)
1. BUBBLE SORT	00:54:14	00:54:24	10
2. SELECTION SORT	00:58:48	00:58:52	4
3. QUICK SORT	00:59:49	00:59:51	2

หมายเหตุ คอมพิวเตอร์ที่ใช้ทดสอบ Pentium Pro200 MHz, Ram 64 MB, HD 6.2 GB ทดสอบการใช้งานการเรียงลำดับข้อมูลในสภาพแวดล้อมของ Visual BASIC (การงานจริงอาจใช้เวลาน้อยกว่านี้มาก)

ในการเรียงลำดับข้อมูล เราจะยังสรุปไม่ได้ว่า การใช้เทคนิค Quick Sort จะเป็นการเรียงลำดับข้อมูลที่เร็วที่สุดหรือดีที่สุด เนื่องจากจะมีปัจจัยหลายประการในการเรียงลำดับข้อมูล เช่น

- จำนวนข้อมูล
- ลักษณะการกระจายของข้อมูล
- ข้อมูลที่มีการเรียงลำดับบางแล้วในบางส่วน

แต่ในการเรียงลำดับข้อมูลที่เป็นข้อมูลชุดเดียวกัน เวลาจะเป็นส่วนสำคัญที่สุดในการพิจารณาว่าอัลกอริธึมใดจะดีที่สุดในการเรียงลำดับข้อมูล นอกจากอัลกอริธึมที่กล่าวมาแล้วยังมีอัลกอริธึมอื่นอีก ได้แก่ Merge Sort, Radix Sort และ Heap Sort เป็นต้น

หวังว่าฟังก์ชันการเรียงลำดับข้อมูลคงจะเป็นประโยชน์กับท่านผู้อ่านทุกท่าน โค้ดการเรียงลำดับข้อมูล ผู้เขียนได้แนวความคิดและได้ดัดแปลงมาจากโค้ดตัวอย่างในภาษา C++ ของ Borland C++ Builder ครับ หากสนใจเป็นภาษา C ก็สามารถไปหาดูได้จากตัวอย่างใน Borland C++ Builder และลองดัดแปลงโค้ดดังกล่าวเพื่อนำมาเขียนเป็น DLL (Dianmic-Link-Library) ดูก็ดีนะครับ ส่วนวิธีการเขียนนั้นสามารถดูได้จากบทความของผู้เขียนที่ผ่านมา ซึ่งจะมีการแนะนำการเขียน DLL และตัวอย่างด้วย หากสงสัยหรือจะแนะนำให้เขียนบทความเกี่ยวกับอะไรก็แนะนำมาได้ที่ VLinux@hotmail.com แล้วพบกันใหม่ สวัสดีครับ หากสนใจต้องการซอร์สโค้ดก็สามารถติดต่อขอใช้บริการได้จากซอฟต์เมล์เช่นเคยครับ

"เรียงลำดับจิตใจของคุณให้เป็นลำดับ เพื่อลดความยุ่งเหยิงของชีวิต"